A picture containing logo

Description automatically generated

Khoa Công nghệ Thông tin

**Midterm project:**

**Xây dựng thiết kế mô hình**

**Hệ thống tập tin.**

Bộ môn An toàn và Phục hồi Dữ liệu

A picture containing outdoor, sign

Description automatically generatedGiảng viên

ThS. Thái Hùng Văn

Sinh viên

19120659 – Phạm Văn Thành

20120382 – Hoàng Thu Thủy

MỤC LỤC

[1 Đánh giá kết quả 3](#_Toc151499039)

[1.1 Đánh giá thành viên 3](#_Toc151499040)

[1.2 Đánh giá mức độ hoàn thành 4](#_Toc151499041)

[2 Nội dung chi tiết 6](#_Toc151499042)

[2.1 Phần lý thuyết 6](#_Toc151499043)

[2.1.1 Chi tiết thiết kế hệ thống tập tin 6](#_Toc151499044)

[2.1.2 Khả năng đáp ứng yêu cầu của thiết kế 10](#_Toc151499045)

[2.2 Phần thực hành 13](#_Toc151499046)

[2.2.1 Tạo / định dạng volume MyFS.DRS 14](#_Toc151499047)

[2.2.2 Thiết lập / Đổi mật khẩu truy xuất MyFS 16](#_Toc151499048)

[2.2.3 Chép (Import) 1 tập tin từ bên ngoài vào MyFS 18](#_Toc151499049)

[2.2.4 Thiết lập / Đổi mật khẩu truy xuất cho 1 tập tin trong MyFS 22](#_Toc151499050)

[2.2.5 Chép (Outport) 1 tập tin trong MyFS ra ngoài 24](#_Toc151499051)

[2.2.6 Liệt kê danh sách các tập tin trong MyFS 25](#_Toc151499052)

[2.2.7 Xóa 1 tập tin trong MyFS 26](#_Toc151499053)

[3 Danh mục tham khảo 29](#_Toc151499054)

# Đánh giá kết quả

## Đánh giá thành viên

|  |  |  |  |
| --- | --- | --- | --- |
| Bảng phân công công việc | | | |
| MSSV | Họ và tên | Công việc | Mức độ đóng góp |
| 19120659 | Phạm Văn Thành | – Xây dựng mô hình và thiết kế kiến trúc tổ chức cho một hệ thống tập tin  – Đặt / Đổi mật khẩu truy xuất cho 1 tập tin trong MyFS  – Chép (Import) 1 tập tin từ bên ngoài vào MyFS  – Chép (Outport) 1 tập tin trong MyFS ra ngoài  – Xóa 1 tập tin trong MyFS | 50% |
| 20120382 | Hoàng Thu Thủy | – Xây dựng mô hình và thiết kế kiến trúc tổ chức cho một hệ thống tập tin  – Tạo / Định dạng volume MyFS.DRS  – Thiết lập / Đổi / Kiểm tra mật khẩu truy xuất MyFS  – Liệt kê danh sách các tập tin trong MyFS | 50% |

## Đánh giá mức độ hoàn thành

|  |  |  |  |
| --- | --- | --- | --- |
| Bảng đánh giá mức độ hoàn thành MÔ HÌNH LÝ THUYẾT | | | |
| Tiêu chí | Tên chức năng | Mức độ hoàn thành | Hạn chế |
| L-1 | Bảo mật thông tin (phòng tránh bị lộ nội dung 1 số tập tin quan trọng) | 100% |  |
| L-2 | An toàn dữ liệu (phòng tránh hư hỏng / mất mát nội dung các tập tin quan trọng) | 100% |  |
| L-3 | Phục hồi dữ liệu đã xóa theo hình thức xóa bình thường có tính khả thi cao và tuyệt đối không thể phục hồi dữ liệu khi đã xóa theo hình thức xóa không cho phục hồi | 100% |  |
| L-4 | Mã hóa dữ liệu (với các dữ liệu cần bảo mật) | 100% |  |
| L-5 | Số tập tin cần tổ chức trong MyFS.DRS có thể đủ nhiều / đa dạng để phải tổ chức hệ thống thư mục phân cấp | 100% |  |
| L-6 | Nội dung dữ liệu bên trong file MyFS.DRS cần được truy xuất theo từng cụm 512 byte | 100% |  |
| L-7 | Bên trong file MyFS.DRS cần có 1 vùng dành riêng để backup các nội dung quan trọng | 100% |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Bảng đánh giá mức độ hoàn thành MÔ HÌNH tHỰC HÀNH | | | |
| Tiêu chí | Tên chức năng | Mức độ hoàn thành | Hạn chế |
| T-1 | Tạo / Định dạng volume MyFS.DRS | 100% |  |
| T-2 | Thiết lập / Đổi / Kiểm tra mật khẩu truy xuất MyFS | 100% |  |
| T-3 | Liệt kê danh sách các tập tin trong MyFS | 90% | Chưa hoàn thành được truy xuất cho thư mục con. |
| T-4 | Đặt / Đổi mật khẩu truy xuất cho 1 tập tin trong MyFS | 100% |  |
| T-5 | Chép (Import) 1 file từ bên ngoài vào MyFS | 100% |  |
| T-6 | Chép (Outport) 1 file trong MyFS ra ngoài | 100% |  |
| T-7 | Xóa 1 tập tin trong MyFS | 100% |  |

# Nội dung chi tiết

## Phần lý thuyết

### Chi tiết thiết kế hệ thống tập tin

**Tổng quát thiết kế:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| VOLUME INFO | ENTRY TABLE | | | | | | DATA TABLE | | | | | |
| ENTRY TABLE 1 | | | ENTRY TABLE 2 | | |
| Main Data | | | Backup Data | | |
| 48 \* 2 = 96 bytes | Entry 0 | … | Entry 254 | Entry 0 | … | Entry 254 | Data block 1 | … | Data block n | Data block\_b 1 | … | Data block\_b n |
| 48 \* 255 bytes | | | 48 \* 255 bytes | | | 512 \* n bytes | | | 512 \* n bytes | | |

* Mỗi *entry* = 48 bytes, mỗi *data block* = 512 bytes.
* VOLUME INFO dùng 2 entries để lưu trữ
* Mỗi Entry Table gồm 255 entries. Tổng kích thước vùng ENTRY TABLE này là 24KB => kích thước của volume phải lớn hơn 24KB.
* Kích thước Data Table = kích thước volume – 24.5 (KB).

**Mô tả chi tiết từng vùng:**

1. **Vùng VOLUME INFO**

|  |  |  |
| --- | --- | --- |
| **VOLUME INFO** | | |
| Địa chỉ | Kích thước (bytes) | Mô tả |
| 0 | 2 | Volume signature: 0x6666 |
| 2 | 16 | Password của volume. Nếu volume không mã hóa, vùng này = 0x00 |
| 18 | 2 | Trạng thái của entry bị xóa sớm nhất |
| 20 | 4 | Kích thước của volume |
| 24 | 4 | Trạng thái của block bị xóa sớm nhất |
| 28 | 4 | Trạng thái của block tiếp theo sẽ xóa |
| 32 | 2 | Trạng thái của entry tiếp theo sẽ xóa |
| 34 | 62 | Vùng để dành cho mã hóa volume (vì mã hóa có gây nhiễu sẽ làm kích thước entry table + data table lớn hơn, khi đó phần dôi ra có thể ghi vào vùng này) |

1. **Vùng ENTRY TABLE**

* Gồm 2 bảng Entry Table:

+ Entry Table 1 là bảng chính,

+ Entry Table 2 là bản backup.

* Có thể linh động thay đổi vai trò tùy trường hợp
* Mỗi entry table chứa tổng cộng 255 entries. Mỗi entry mô tả một file/folder, có kích thước cố định cho một entry là 48 bytes.

|  |  |  |
| --- | --- | --- |
| **ENTRY** | | |
| Địa chỉ | Kích thước (bytes) | Mô tả |
| 0 | 2 | Trạng thái của entry   * 1: Đang dùng * 0: Chưa dùng * 2 - 256: Đã xóa. Dựa vào “Trạng thái của entry bị xóa sớm nhất” đã lưu trữ ở *Volume Info* để biết Entry nào đã được xóa sớm nhất. |
| 2 | 1 | Entry này mô tả tập tin hay thư mục   * 10h: Thư mục * 20h: Tập tin |
| 3 | 1 | Số thứ tự của entry thư mục cha. Nếu đây là file/folder ngoài cùng, giá trị này = 0. |
| 4 | 4 | Block đầu tiên của file trên vùng data. Nếu entry mô tả thư mục, giá trị này = 0xFFFFFFFF. |
| 8 | 4 | Kích thước file. Biểu diễn được lớn nhất là 2^32, vậy nên chỉ quản lý được file có dung lượng lớn nhất là 4GB. Nếu entry mô tả thư mục, giá trị này = 0. |
| 12 | 16 | Password của file. Nếu file không mã hóa, vùng này = 0. |
| 28 | 16 | Tên file/folder. Do đó tên file/folder chỉ có tối đa 16 kí tự. |
| 44 | 4 | Checksum của file. |

1. **Vùng DATA TABLE**

**\*) Main data:**

* Vùng này gồm nhiều block, mỗi block có kích thước 512 bytes. Tùy vào kích thước mỗi file mà sẽ được cấp phát số lượng block phù hợp để lưu trữ.
* Cấp phát số lượng block phù hợp: Không phải chỉ cấp phát vừa đủ để lưu nội dung file, mà cấp phát thừa ra một khoảng. Mục đích là để khi mã hóa sẽ làm kích thước file lớn hơn, khi đó sẽ có đủ chỗ để ghi file.

**\*) Back up data:**

- Vùng này đơn giản chỉ copy lại data từ vùng main data phòng trường hợp data bên main bị hỏng,

- Các dữ liệu quan trọng có thể đem back up lại từ vùng này

|  |  |  |
| --- | --- | --- |
| **DATA\_BLOCK** | | |
| Địa chỉ | Kích thước (bytes) | Mô tả |
| 0 | 4 | Trạng thái của block   * 1: Đang dùng * 0: Chưa dùng * 2 – 4 000 000 000: Đã xóa. Dựa vào “Trạng thái của block bị xóa sớm nhất” đã lưu trữ ở Volume Info để biết block nào đã được xóa sớm nhất. |
| 4 | 4 | Địa chỉ của block tiếp theo. Nếu đây là block cuối của file, giá trị vùng này là 0xFFFFFFFF. |
| 8 | 504 | Nội dung file |

### Khả năng đáp ứng yêu cầu của thiết kế

1. Việc bảo mật thông tin (phòng tránh bị lộ nội dung 1 số tập tin quan trọng) được xem là thiết yếu nhất.

Đáp ứng: 100%.

Lí do: Thiết kế trên cho phép mã hóa Volume và cho phép mã hóa từng file cụ thể, cũng có thể mã hóa cả folder.

1. Việc an toàn dữ liệu (phòng tránh hư hỏng /mất mát nội dung các tập tin quan trọng) cũng rất cần thiết.

Đáp ứng: 100%.

Lí do: Thiết kế trên có 2 bảng ENTRY TABLE. Nếu có hư hỏng 1 trong 2 bảng thì vẫn còn bản backup. Vùng DATA TABLE cũng được chia thành khu vực lưu trữ chính và khu vực lưu trữ dự phòng giúp phòng tránh hư hỏng hay mất mát nội dung quan trọng.

1. Việc phục hồi dữ liệu đã xóa theo hình thức xóa bình thường có tính khả thi cao (thiết kế để hạn chế việc chúng bị chép đè), và tuyệt đối không thể phục hồi dữ liệu khi đã xóa theo hình thức xóa không cho phục hồi.

Đáp ứng: 100%.

Thiết kế trên lưu dữ liệu file theo từng cụm bytes - block. Ví dụ: File có dung lượng 1024 bytes sẽ được lưu trong khoảng 3 blocks (512\*3 = 1536), do 1024 bytes + số lượng byte dự trữ cho việc mã hóa (vì mã hóa sẽ làm tăng kích thước file do có thêm chuỗi gây nhiễu). Ngoài ra, 3 blocks này có thể không nằm liền kề nhau (trên thiết kế DATA có mục Địa chỉ của block tiếp theo).

Việc lưu theo block khiến chương trình chạy nhanh hơn khi tìm kiếm block trống để lưu trữ (vì dò theo cụm 512 bytes) so với thiết kế lưu data sát bên nhau (phải dò theo từng byte để tìm khoảng trống).

Quan trọng nhất, thiết kế thỏa mãn yêu cần hạn chế bị chép đè:

– Khi xóa file/folder (xóa thường):

* Dò trên Volume Info tìm xem entry bị xóa tiếp theo đặt là mấy (mục: Trạng thái của entry tiếp theo sẽ xóa).
* Gán giá trị này cho trạng thái của entry cần xóa.
* Tăng giá trị của Trạng thái của entry tiếp theo sẽ xóa lên 1. Nếu giá trị này đang là 256, đặt lại là 2. Vì vùng này chỉ có 255 entries nên ta chỉ cần khoảng giá trị 2 – 256 là có thể đánh dấu được toàn bộ 255 entries.

– Khi thêm file/folder:

* Dò tìm entry có trạng thái là 0 (nghĩa là entry chưa được dùng bao giờ, hoặc đã được xóa hoàn toàn). Nếu có, ghi lên entry này.
* Nếu không có, dò tìm entry bị xóa sớm nhất (mục: Trạng thái của entry bị xóa sớm nhất). Nếu có, ghi đè entry này và tăng giá trị Trạng thái của entry bị xóa sớm nhất lên 1. Nếu giá trị này đang là 256, đặt lại là 2. Vì vùng này chỉ có 255 entry nên ta chỉ cần khoảng giá trị 2 – 256 là có thể đánh dấu được toàn bộ 255 entry.
* Nếu vẫn không có, nghĩa là đã dùng hết 255 entry.

– Khi xóa file/folder không phục hồi: Đặt tất cả các bytes của entry và các block tương ứng của nó về trạng thái ban đầu 0x00. Điều này khiến file/folder mất dữ liệu hoàn toàn và không thể phục hồi.

1. Việc mã hóa dữ liệu (với các dữ liệu cần bảo mật) phải đáp ứng các tiêu chí:

– Mã theo khóa là mật khẩu được thiết lập riêng biệt cho từng tập tin.

– Không lưu lại mật khẩu.

– Với cùng 1 nội dung và mật khẩu, mỗi lần mã hóa sẽ cho ra 1 bản mã có nội dung và chiều dài khác nhau.

Đáp ứng: 100%.

Lí do:

– Mỗi entry có vùng để lưu mật khẩu đã hash MD5 (128bit), cho nên có thể mã hóa riêng biệt từng tập tin.

– Mã hóa EAS cho nội dung từng tập tin.

– Thiết kế cấp phát thừa ra một khoảng trong vùng data, đáp ứng được mã hóa cho ra bản mã kích thước khác nhau.

1. Số tập tin cần tổ chức trong MyFS.DRS có thể đủ nhiều / đa dạng để phải tổ chức hệ thống thư mục phân cấp.

Đáp ứng: 100%.

Lí do: Với 255 entries cho một ENTRY TABLE và mỗi Entry mô tả một tập tin/ folder thì số lượng tập tin + thư mục trong thiết kế trên tối đa là 255. Thực tế, phần ENTRY TABLE này có thể quy định nhiều hơn hoặc ít hơn. Người dung có thể chỉ định số lượng Entry hay để chương trình tự đề xuất một lượng phù hợp tùy theo dung lượng volume được chỉ định.

1. Nội dung dữ liệu bên trong file MyFS.DRS cần được truy xuất theo từng cụm 512 byte giống như sector trên các volume của HĐH hoặc các đĩa của máy tính. (tức cần thiết kế & xây dựng các hàm ReadBlock / WriteBlock giống các hàm ReadSector / WriteSector của HĐH & máy tính).

Đáp ứng: 100%.

Lí do:

– Người dùng chỉ định kích thước volume với đơn vị KB, do đó luôn chia hết cho 512.

– Kích thước của phần Volume Info + Entry Table chia hết cho 512, dễ dàng quản lí theo block.

– Phần data được chia thành các block, do đó cũng dễ dàng quản lí theo block.

1. **Bên trong file MyFS.DRS cần có 1 vùng dành riêng để backup các nội dung quan trọng.**

Đáp ứng: 100%.

Lí do:

Bảng DATA TABLE đã có một vùng dành riêng (BACKUP DATA) để sao lưu nội dung trong MAIN DATA điều này cho phép backup dữ liệu khi cần thiết nếu nội dung tập tin bị hư hỏng / mất mát.

## Phần thực hành

* Phần này có những điểm tinh giản so với thiết kế mô hình lý thuyết như:
* Chỉ có 1 bảng Entry Table để dễ dàng cài đặt.
* Số lượng entry trong Entry Table cũng giới hạn là 30 để dễ dàng kiểm thử chức năng hạn chế chép đè. Phần này có thể thay đổi: #define NUM\_ENTRY 30 trong file “Volume.h”.
* Entry trong Entry Table không có checksum, thay vào đó là tên file có độ dài tối đa 20 kí tự.

- Người dung có thể sử dụng `help` để xem cách dùng các lệnh:

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

### Tạo / định dạng volume MyFS.DRS

– Tạo volume có tên myfs.drs với kích thước 128KB:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động

Volume được tạo thành công:

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

Khi vừa khởi tạo, tất cả các bytes đều là 0, trừ một số vùng như:

* Signature: 0x6666
* Entry xóa đầu tiên / Entry xóa tiếp theo: 0x02, 0x02
* Kích thước volume : 0x80 = 128
* Block xóa đầu tiên/Block xóa tiếp theo: 0x02, 0x02

Vùng còn lại là 16 bytes password để chứa mật khẩu sau khi hash, và 62 bytes để lưu trữ trong trường hợp mã hóa volume. Do chưa đặt mật khẩu hay mã hóa nên các vùng này hiện vẫn là các byte 0.

### Thiết lập / Đổi mật khẩu truy xuất MyFS

– Đặt mật khẩu cho volume mới tạo là 123456.

+ Mật khẩu sẽ hash và được cập nhật cho volume:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Mô tả được tạo tự động

– Đổi mật khẩu thành: 19001009

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Mô tả được tạo tự động

Đổi mật khẩu thành công.

* Khi người dùng muốn mở volume, password sẽ được yêu cầu:

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

### Chép (Import) 1 tập tin từ bên ngoài vào MyFS

– Import tập tin có đường dẫn: *C:/Users/Klaus/Desktop/life.txt* từ bên ngoài vào **myfs**.

+ Hệ sống sẽ tự loại bỏ đường dẫn và giữ lại filename: *life.txt*

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, màn hình

Mô tả được tạo tự động

File được thêm vào lúc này sẽ chưa được mã hóa. File này được lưu trong 3 block: 0, 1, 2.

+ Các 4 bytes đầu của các block được đánh dấu là 01 chứng tỏ các block này đang được sử dụng.

+ Block đầu tiên (block\_0) chỉ đến block\_1 (4 bytes được đánh đấu màu xanh lá)

+ Block\_1 chỉ đến block\_2 (4 bytes được đánh đấu màu xanh lá)

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

+ Block \_2 chỉ đến 0xFFFFFFFF, điều này có nghĩa là nó là block cuối cùng của file:

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, số

Mô tả được tạo tự động

Như đã trình bày ở mô hình lý thuyết chương trình có thể tìm các block còn trống để ghi file và các block này không nhất thiết phải ở gần nhau:

* Dưới đây là file **text3.txt** được thêm vào, khi ở đầu vùng DATA do có các block bị xóa vĩnh viễn ở file cũ trước đó.
* Khi import file mới vào, chương trình sẽ ghi vào các block trống trước (**3, 4**), sau đó là các block trống phía sau: **35,36,..**

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

* Trong trường hợp đã hết block trống (ký hiệu 0x00 ở 2 bytes đầu) thì hệ thống sẽ tìm các block của file đã bị xóa tạm thời( không phải vĩnh viễn).

Thuật toán cho phần này giống với phần lý thuyết 2.1.2.3 đã trình bày ở trên.

* Trường Import một file khi volume không đủ kích thước: 160kb

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

### Thiết lập / Đổi mật khẩu truy xuất cho 1 tập tin trong MyFS

– Đặt mật khẩu cho file *life.txt* là : **123456**

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, màn hình

Mô tả được tạo tự động

Đặt mật khẩu cho file thành công. File đã được mã hóa.

Thuật toán mã hóa (hàm encryptFile() trong Volume.h)

* Gây nhiễu bằng cách thêm random các chuỗi gây nhiễu vào nội dung file. Thêm sao cho chiều dài file chia hết cho 16.
* Dùng AES 128 mã hóa nội dung trên.

Ngược lại, để giải mã (hàm decryptFile() trong Volume.h)

* Dùng AES 128 giải mã.
* Loại bỏ các chuỗi gây nhiễu và trả về bản rõ.

– Đổi password cho file *life.txt* thành **19001009**.

* File lúc này sẽ mang nội dung mã hóa khác hoàn toàn khác

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

### Chép (Outport) 1 tập tin trong MyFS ra ngoài

– Outport tập tin *life.txt* đã đặt mật khẩu 123456 ra ngoài với tên là *life\_out.txt*.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, phần mềm

Mô tả được tạo tự động

File được giải mã và outport thành công:

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

### Liệt kê danh sách các tập tin trong MyFS

– Import thêm file: *text1.txt* và *text2.txt* vào volume.

Kết quả khi chạy dòng lệnh iệt kê danh sách tập tin: **`ls`**

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, máy tính

Mô tả được tạo tự động

### Xóa 1 tập tin trong MyFS

1. **Xóa thường**

* Xóa thường file *life.txt*

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, đa phương tiện

Mô tả được tạo tự động

Entry và block của file này vẫn còn, chỉ bị đánh dấu sang trạng thái khác- đã xóa kèm số thứ tự

* Hiện tại đây là file bị xóa đầu tiên nên sẽ bắt đầu từ số 2.

+ xóa trên entry:

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

+ xóa trên data\_block:

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

1. **Xóa không phục hồi**

– Xóa không phục hồi file: *text1.txt*

Để xóa không phục hồi, dùng lệnh rm <tên file> true hoặc rm <tên file> 1

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, màn hình

Mô tả được tạo tự động

Thông tin trên entry và block liên quan tới file này hoàn toàn biến mất và các byte được set lại thành 00.

# Danh mục tham khảo

– C++ md5 function, zedwood.com, <http://www.zedwood.com/article/cpp-md5-function>

– GitHub Reponsitory ‘AES’, Tác giả ‘SergeyBel’, <https://github.com/SergeyBel/AES>